

IAP5 Rec'd PCT/PTO 29 MAR 2006

## Beschreibung

## Überprüfung der Verfügbarkeit eines Servers

- 5 Die vorliegende Erfindung betrifft ein Verfahren zur Überprüfung der Verfügbarkeit eines Servers, ein Steuerungsprogramm und einen Client für ein verbindungslose Dienste bereitstellendes Netz.
- 10 Eine zuverlässige Überprüfung der Verfügbarkeit eines Servers ist insbesondere bei lose gekoppelten Client-Server-Beziehungen in paketorientierten Datennetzen von großer Bedeutung. Erkennt ein Client frühzeitig, daß ein Server überlastet oder
- 15 ausgefallen ist, so können noch rechtzeitig Gegenmaßnahmen eingeleitet werden, beispielsweise Suche nach einem Alternativ-Server oder Erzeugen von Warnhinweisen. Verfahren zur Überprüfung der Verfügbarkeit eines Servers, zu denen auch im H.323-Standard, Stand 11/2000, Kap. 7.2.2 beschriebene Keep-Alive-Tests zählen, werden angewendet, wenn keine permanente
- 20 Kommunikationsbeziehung zwischen Client und Server besteht, ein fehlerfreies Bestehen einer solchen Beziehung jedoch Grundlage für eine gewünschte Funktionalität ist, beispielsweise Internet-Telephonie. In paketorientierten Netzen dienen Keep-Alive-Tests beispielsweise dazu, Kommunikationsteilnehmern einen quasi leitungsvermittelnden Charakter hinsichtlich
- 25 gegenseitiger Erreichbarkeit zu simulieren.

- Bei gängigen Keep-Alive-Tests sendet ein Client in zyklischen Zeitabständen Verfügbarkeitsanfragen an einen ausgewählten
- 30 Server. Wird eine Verfügbarkeitsanfrage durch den Server innerhalb eines vorgebbaren Zeitraums beantwortet, gilt der Server als verfügbar und die Kommunikationsbeziehung daher als aktiv. Nachteilig an dieser Vorgehensweise ist, daß der Server in der Regel durch eine Beantwortung sämtlicher
- 35 Client-Anfragen erheblich belastet wird.

## 2

Der vorliegenden Erfindung liegt daher die Aufgabe zugrunde, ein Verfahren zur Überprüfung der Verfügbarkeit eines Servers, das eine Minimierung der Belastung des Servers durch an ihn gerichtete Verfügbarkeitsanfragen ermöglicht, sowie zur  
5 Durchführung des Verfahrens geeignete technische Implementierungen anzugeben.

Diese Aufgabe wird erfindungsgemäß durch ein Verfahren mit den in Anspruch 1, ein Steuerungsprogramm mit den in Anspruch  
10 7 und einen Client mit den in Anspruch 8 angegebenen Merkmalen gelöst. Vorteilhafte Weiterbildungen der vorliegenden Erfindung sind in den abhängigen Ansprüchen angegeben.

Ein wesentlicher Aspekt der vorliegenden Erfindung besteht  
15 darin, daß ein Client, der auf eine an einen Server gerichtete Verfügbarkeitsanfrage eine Bestätigungsmeldung vom Server empfangen hat, eine Meldung über die Verfügbarkeit des Servers an weitere Clients übermittelt. Daraufhin unterbinden  
20 die vorgebbaren weiteren Clients zumindest für ein vorgebbares Zeitintervall ein Übermitteln einer Verfügbarkeitsanfrage an den Server. Auf diese Weise kann die Belastung des Servers durch Verfügbarkeitsanfragen erheblich reduziert werden, ohne daß dies beispielsweise zu Lasten einer nicht mehr rechtzeitigen Erkennung eines Serverausfalls oder einer Serverüber-  
25 lastung geht.

Die vorliegende Erfindung wird nachfolgend an einem Ausführungsbeispiel anhand der Zeichnung näher erläutert. Es zeigt

30 Figur 1 ein Anwendungsumfeld der vorliegenden Erfindung mit zahlreichen Clients und Servern in einem paketorientierten Netz mit mehreren Teilnetzen,

Figur 2 ein Ablaufdiagramm für ein Verfahren zur Überprüfung der Verfügbarkeit eines Servers,  
35

Figur 3 ein Ablaufdiagramm für ein gegenüber Figur 2 modifiziertes Verfahren, bei dem zusätzlich die Verfügbarkeit von Clients durch den Server überprüft wird.

5

In Figur 1 ist ein Kommunikationssystem dargestellt, das ein paketorientiertes Netz mit mehreren Teilnetzen 101, 110, 120, 130, Server 102 bis 104 und mehrere Clients 111 bis 115, 121 bis 123, 131 bis 133 umfaßt. Die Clients 111 bis 115, 121 bis 123, 131 bis 133 nutzen von den Servern 102 bis 104 angebotene Dienste, beispielsweise Internet-Telefonie.

Bei den Teilnetzen 110, 120, 130 handelt es sich um lokale Subnetze, die jeweils über einen Gatekeeper verfügen. Die Gatekeeper dienen vornehmlich zur Zugangskontrolle und zu Adreßübersetzungsdiensten. Diese Funktionalität der Gatekeeper kann durch Server im paketorientierten Netz übernommen werden.

Zur Nutzung von durch einen Server 102 bis 104 angebotenen Diensten registrieren sich die Clients 111 bis 115, 121 bis 123, 131 bis 133 zunächst am jeweiligen Server. Dabei wird ein Zeitraum spezifiziert, innerhalb dessen eine Registrierung als gültig behandelt wird. Der Ablauf eines Registrierungsverfahrens wird exemplarisch und ohne Beschränkung der Allgemeingültigkeit für den Client 111 dargestellt.

Der Client 111 verfügt über eine zentrale Prozessoreinheit (CPU), einen Arbeitsspeicher (RAM), eine Festplatte zur nichtflüchtigen Speicherung von Daten (HD), einen Netzwerkadapter (Tx/Rx) und eine Zeitsteuerungseinheit (Timer). Einen solchen Aufbau und eine solche Funktionalität können auch die übrigen in Figur 1 dargestellten Clients 112 bis 115, 121 bis 123, 131 bis 133 aufweisen.

35

Zur Registrierung an einem der Server 102 bis 104 übermittelt der Client 111 eine Meldung mit einer Registrierungsanforde-

5      rung RRQ an einen Server, der einen gewünschten Dienst bereitstellt. In der Meldung mit der Registrierungsanforderung RRQ wird ein Zeitraum spezifiziert, innerhalb dessen die Registrierung des Clients 111 am ausgewählten Server gültig sein soll. Außerhalb dieses Zeitraums gilt die Registrierung des Clients als ungültig.

10      Bei Verfügbarkeit antwortet der ausgewählte Server durch eine Meldung mit Registrierungsbestätigung RCF auf die Registrierungsanfrage. In der Meldung mit der Registrierungsbestätigung RCF ebenfalls einen Gültigkeitszeitraum für die Registrierung spezifiziert. Der vom ausgewählten Server bestätigte Gültigkeitszeitraum entspricht entweder dem vom Client 111 angegebenen Zeitraum oder ist kürzer als dieser.

15      Innerhalb des Gültigkeitszeitraums für die Registrierung kann der Client 111 Verfügbarkeitsanfragen an den ausgewählten Server richten. Derartige Verfügbarkeitsanfragen, die auch als Keep-Alive-Tests bezeichnet werden, werden ebenfalls in Form einer Meldung mit einer Registrierungsanfrage RRQ an den ausgewählten Server übermittelt. Eine im Rahmen eines Keep-Alive-Tests übermittelte Meldung mit Registrierungsanfrage RRQ weist gegenüber gewöhnlichen Meldungen mit Registrierungsanfrage ein gesetztes Keep-Alive-Bit ist.

25      Entsprechend den ITU-T-Empfehlungen H.225.0, Stand 11/2000, Kapitel 7.9 ist eine verkürzte Meldung mit Registrierungsanfrage für eine "Lightweight Registration" bereits ausreichend. Dabei weist die Meldung mit Registrierungsanfrage lediglich Angaben über eine Registrierungs- und Status-Transportadresse (RAS-Adresse) des Clients, ein gesetztes Keep-Alive-Bit, einen Endpunktidentifikator für den Client, einen Server- bzw. Gatekeeperidentifikator, Berechtigungsmarken (Tokens) für auszuführende Operationen und einen Gültigkeitszeitraum für die Registrierung.

30     

35

Eine Meldung mit Registrierungsanforderung RRQ, die im Rahmen eines Keep-Alive-Tests an den ausgewählten Server übermittelt wird, kann auch zur Verlängerung des Gültigkeitszeitraums der Registrierung verwendet werden. Dies gilt jedoch nur dann, wenn die Meldung noch innerhalb des Gültigkeitszeitraums der Registrierung vom ausgewählten Server empfangen wird. Daher sollten zur Ermittlung des Endes eines Gültigkeitszeitraums durch einen Client auch Meldungs- und Bearbeitungsverzögerungen berücksichtigt werden. Nach dem Ende des Gültigkeitszeitraums der Registrierung ist zur erneuten Registrierung eine Übermittlung einer oben beschriebenen verkürzten Meldung mit Registrierungsanfrage nicht mehr ausreichend.

Falls die Meldung mit der Registrierungsbestätigung RCF keine Angabe über einen Gültigkeitszeitraum der Registrierung aufweist, wird dies dahingehend gewertet werden, daß der ausgewählte Server keine Keep-Alive-Mechanismen für Verfügbarkeitsanfragen unterstützt. Clients sollten bei Empfang einer solchen Meldung ohne Angabe über einen Gültigkeitszeitraum ein Versenden von Verfügbarkeitsanfragen an den jeweiligen Server unterbinden.

Nach Ablauf der Gültigkeit einer Registrierung kann der ausgewählte Server eine Meldung mit einem entsprechenden Hinweis an den betroffenen Client übermitteln, um diesen über den Ablauf des Gültigkeitszeitraums zu informieren. Dies ist insbesondere bei einem Verlust der Synchronisierung zwischen Zeitsteuerungseinheiten am Client einerseits und am Server andererseits von Vorteil.

Sendet der Client 111 nach dem Ende des Gültigkeitszeitraums der Registrierung eine oben beschrieben verkürzte Meldung mit Registrierungsanfrage RRQ und gesetztem Keep-Alive-Bit an den ausgewählten Server, so wird der Server durch eine Meldung mit Registrierungs zurückweisung RRJ reagieren. In einer solchen Meldung kann auch ein Zurückweisungsgrund angegeben sein.

Um eine unnötige Belastung eines Servers 102 bis 104 durch Verfügbarkeitsanfragen von Clients 111 bis 115, 121 bis 123, 131 bis 133 zu vermeiden, informiert ein Client, der eine positive oder eine negative Antwort auf eine Verfügbarkeitsanfrage erhalten hat, mittels einer Multicast-Meldung MCM sämtliche Clients innerhalb desselben Teilnetzes 110, 120, 130 über die Verfügbarkeit des jeweiligen Servers. Durch eine Begrenzung auf dasselbe Teilnetz können außerdem Probleme im Hinblick auf nicht allgemein auflösbare Adressen umgangen werden. Insbesondere wird eine unnötige Netzbelastung durch nichtadressierbare Meldungen verhindert.

Das in Figur 2 dargestellte Ablaufdiagramm dient zur weiteren Veranschaulichung eines Verfahrens zur Überprüfung der Verfügbarkeit eines Servers in einem paketerorientierten Kommunikationsnetz. Ausgangspunkt des Verfahrens ist ein Start eines Timers bzw. eines Zählers einer Zeitsteuerungseinheit eines Clients (Schritt 201). Durch den Timer werden Zeitpunkte festgelegt, zu denen Verfügbarkeitsanfragen des Clients an einen ausgewählten Server übermittelt werden sollen.

Nach dem Start des Timers überwacht der Client den Empfang von Multicast-Meldungen MCM, aus denen sich Aussagen über die Verfügbarkeit eines Servers ergeben (Schritt 202). Empfängt der Client eine solche Multicast-Meldung, so wird der Timer zurückgesetzt (Schritt 208). Dadurch wird eine Übermittlung von Verfügbarkeitsanfragen an den ausgewählten Server für ein vorgebbares Zeitintervall unterbunden.

30  
35  
Sofern keine neue Multicast-Meldung empfangen wurde, wird zusätzlich überprüft, ob der Timer abgelaufen und damit ein Zeitpunkt zum Absetzen einer neuen Verfügbarkeitsanfrage erreicht ist (Schritt 203). Wenn der Timer noch nicht abgelaufen ist, werden weiterhin der Empfang von Multicast-Meldungen und der Ablauf des Timers überprüft. Ist der Timer abgelau-

fen, wird eine Verfügbarkeitsanfrage an den ausgewählten Server versendet (Schritt 204).

In Schritt 205 wird zumindest implizit die Verfügbarkeit oder  
5 Nichtverfügbarkeit des ausgewählten Servers festgestellt. Ist  
der ausgewählte Server verfügbar, so antwortet dieser durch  
eine an den anfragenden Client gerichtete Bestätigungsmeldung  
auf die Verfügbarkeitsanfrage (Schritt 206). Nach Erhalt der  
Bestätigungsmeldung informiert der anfragende Client sämtli-  
10 che weiteren Clients innerhalb desselben Teilnetzes über die  
Verfügbarkeit des ausgewählten Servers mittels einer Multi-  
cast-Meldung (Schritt 207). Daraufhin unterbinden die weite-  
ren Clients, welche die Multicast-Meldung empfangen haben,  
ihrerseits ein Übermitteln von Verfügbarkeitsanfragen an den  
15 ausgewählten Server für ein vorgegebenes Zeitintervall.

Da bei Ausfall oder Überlastung des ausgewählten Servers kei-  
ne Bestätigungsmeldung über dessen Verfügbarkeit versendet  
wird, wird überprüft, ob der ausgewählte Server zumindest in  
20 der Lage ist, mit einer Nichtverfügbarkeitsmeldung auf die  
Verfügbarkeitsanfrage zu reagieren (Schritt 209). Bei einer  
Überlastung des ausgewählten Servers wäre dies beispielsweise  
noch möglich. Dagegen ist bei einem Ausfall des ausgewählten  
Servers nicht mehr mit einem Versenden einer Nichtverfügbar-  
25 keitsmeldung zu rechnen. Daher wird überprüft ob ein Zeitli-  
mit für den Empfang einer Bestätigungsmeldung oder einer  
Nichtverfügbarkeitsmeldung erreicht wird (Schritt 211). Wird  
dieses Zeitlimit überschritten, so übermittelt der anfragende  
Client eine Multicast-Meldung über die Nichtverfügbarkeit des  
30 ausgewählten Servers an sämtliche Clients innerhalb desselben  
Teilnetzes (Schritt 210). In entsprechender Weise wird ver-  
fahren, wenn der anfragende Client innerhalb des für eine An-  
wort auf die Verfügbarkeitsanfrage zulässigen Zeitlimits eine  
Nichtverfügbarkeitsmeldung erhält.

35

Das beschriebene Verfahren zur Überprüfung der Verfügbarkeit  
eines Servers wird vorzugsweise durch ein Steuerungsprogramm

implementiert, das in einen Arbeitsspeicher eines Clients ladbar ist und zumindest einen Code-Abschnitt aufweist, bei dessen Ausführung die vorangehend beschriebenen Schritte Übermitteln der Verfügbarkeitsanfrage, Überwachen des Empfangs einer Bestätigungsmeldung, Übermitteln einer Meldung über die Verfügbarkeit an weitere Clients, Überwachen des Empfangs von Meldungen weiterer Clients und ggf. Unterbinden des Versendens von Verfügbarkeitsanfragen ablaufen. Mittels des Steuerungsprogramms wird ein Client für ein verbindungslose Dienste bereitstellendes Kommunikationsnetz implementiert, der eine Einrichtung zum Übermitteln von Verfügbarkeitsanfragen, eine Einrichtung zur Überwachung des Empfangs von Bestätigungsmeldungen, eine Einrichtung zur Übermittlung von Meldungen über die Verfügbarkeit von Servern an weitere Clients sowie eine Einrichtung zur Überwachung des Empfangs von Meldungen weiterer Clients und zum Unterbinden des Übermittels von Verfügbarkeitsanfragen aufweist.

Die nachfolgenden Betrachtungen dienen einer Herleitung der Vorteile des beschriebenen Verfahrens zur Überprüfung der Verfügbarkeit eines Servers gegenüber bisher gängigen Keep-Alive-Tests. Entsprechend den bisher gängigen Keep-Alive-Tests würde ein Server bei  $n=1000$  Verfügbarkeitsanfragen sendenden Clients und einer Anfragerate von  $a=3$  Anfragen pro Minute und Client sowie unter der Annahme einer zeitlichen Gleichverteilung derartiger Anfragen alle

$$t_r(a=3, n=1000) = \frac{60s}{a \cdot n} = \frac{60s}{3 \cdot 1000} = 20ms$$

eine Verfügbarkeitsanfrage zu beantworten haben.

Der Server wäre also durchschnittlich alle 20 ms damit beschäftigt, eine Verfügbarkeitsanfrage zeitnah zu beantworten. Dies würde zu einer erheblichen Belastung des Servers durch Verfügbarkeitsanfragen führen, was ihn in der Wahrnehmung seiner ihm eigentlich zugedachten Aufgaben stark einschränken würde. Zur Lösung dieses Problems würde sich zunächst anbieten, die Anfragerate der Clients pro Minute stark zu reduzie-



ren, was jedoch zur Folge hätte, daß die betroffenen Clients den Verlust der Verfügbarkeit des Server unter Umständen unzureichend spät feststellen.

- 5 Für eine Abschätzung einer mit dem hier vorgeschlagenen Verfahren erzielbaren Vorteile wird zunächst angenommen, daß sich  $n$  Clients auf  $s$  Teilnetze verteilen. Ferner wird angenommen, daß aus jedem dieser  $s$  Teilnetze wenigstens ein Client versuchen wird, die Verfügbarkeit des Servers abzufragen. Jeder dieser Client würde dann wiederum die übrigen Clients innerhalb seines Teilnetzes mittels einer Multicast-Meldung über die Serververfügbarkeit informieren. Unter idealisierten Bedingungen würde die Anzahl der durch den Server zu bearbeitenden Verfügbarkeitsanfragen der Anzahl  $s$  der Teilnetze entsprechen. Zusätzlich ist noch zu berücksichtigen, daß die Multicast-Meldungen in den Teilnetzen eine Verlustrate  $v$  zwischen 0 und 100 Prozent aufweisen können. Damit ist die Anzahl  $v_c$  von Clients pro Teilnetz, die von einer Multicast-Meldung nicht erreicht werden und somit selber eine Verfügbarkeitsanfrage an den Server senden abhängig von der Verlustrate  $v$  der Multicast-Meldungen:

$$v_c = v \left( \frac{n}{s} - 1 \right).$$

- Die Gesamtzahl  $v_c^{\text{all}}$  aller Clients, die unter dieser Voraussetzung tatsächlich eine Verfügbarkeitsanfrage an den Server richten läßt sich damit wie folgt abschätzen:

$$v_c^{\text{all}} = s \cdot v \cdot \left( \frac{n}{s} - 1 \right) + s = v \cdot n - v \cdot s + s = v \cdot n + s(1 - v).$$

- Unter diesen Annahmen kann das Zeitintervall  $t_r$  zwischen zwei an den Server gerichteten Verfügbarkeitsanfragen wie folgt berechnet werden:

$$t_r(a, n, s, v) = \frac{60s}{a \cdot (v \cdot n + s(1 - v))}.$$

- Anhand eines Zahlenbeispiels wird die Belastung des Servers durch Verfügbarkeitsanfragen bei bisher gängigen Keep-Alive-

10

Tests der Belastung des Servers entsprechend dem hier vorgeschlagenen Verfahrens gegenüber gestellt. Bei einer Anfragerate von  $a=3$  Verfügbarkeitsanfragen pro Minute und  $n=1000$  Clients gilt für das Zeitintervall  $t_r$  zwischen zwei Verfügbarkeitsanfragen:

$$t_r(a=3, n=1000, s, v) = \frac{60s}{3 \cdot (v \cdot 1000 + s(1-v))} = \frac{20s}{1000 \cdot v + s(1-v)}.$$

Typische Werte für die Verlustrate  $v$  liegen deutlich unter 100 Prozent. Mit den obigen Werten für die Anfragerate  $a$ , die Anzahl  $n$  der Clients und  $s=10$  Teilnetzen ergeben sich in Abhängigkeit von der Verlustrate  $v$  die folgenden Effizienzsteigerungen  $d_{eff}$  in Prozent bezogen auf die gängigen Keep-Alive-Tests mit einem Zeitintervall  $t_r = 20$  ms zwischen zwei Verfügbarkeitsanfragen:

$v$	0.0	0.1	0.2	0.5	0.8
$t_r[\text{ms}]$	2000	183	96	40	25
$d_{eff}[\%]$	9900	815	380	100	25

Bei einer Verlustrate  $v = 20\%$  beträgt die mittlere Zeit  $t_r$  zwischen zwei durch den Server zu bearbeitende Verfügbarkeitsanfragen beispielsweise ca. 96 ms. Dies entspricht einer Effizienzsteigerung  $d_{eff} = 380\%$  im Vergleich zu bisher gängigen Keep-Alive-Tests.

Insgesamt ist festzustellen, daß bei dem hier vorgeschlagenen Verfahren zur Überprüfung der Serververfügbarkeit die Belastung des Servers durch eine Beantwortung von Verfügbarkeitsanfragen deutlich sinkt, ohne daß dies zu einer signifikanten Erhöhung der Zeit führt, bis den Clients die Nichtverfügbarkeit des Servers bekannt wird. Außerdem sind zur Implementierung des hier vorgeschlagenen Verfahrens lediglich auf Seiten der Clients Änderungen notwendig. Die bisherige Vorgehensweise zur Bearbeitung von Verfügbarkeitsanfragen kann auf Serverseite beibehalten werden und erfordert damit keine weiteren Aufwendungen.

Bei dem in Figur 3 dargestellten Verfahren wird zusätzlich die Verfügbarkeit von Clients durch einen Server überprüft. Dies bedeutet, daß ein Server auch über die Verfügbarkeit von Clients informiert ist, die lediglich von einem anderen Client per Multicast-Meldung über die Verfügbarkeit des Servers informiert werden und nicht selber Keep-Alive-Anfragen direkt an den Server richten. Zunächst startet jeder Client einen ersten Timer (Timer 1) (Schritt 301) und überprüft, ob eine Multicast-Sammelanfrage empfangen wurde (Schritt 302). Eine Multicast-Sammelanfrage kann von einem Client als Zeichen dafür ausgesendet werden, daß dieser Client einen Sammel-Keep-Alive-Test mit einem Server durchführen möchte, und daß dieser Client dafür die Keep-Alive-Daten anderer Clients benötigt.

Der Zustand des ersten Timers wird fortlaufend durch den jeweiligen Client überprüft (Schritt 303). Sollte der erste Timer noch nicht abgelaufen und keine Multicast-Sammelanfrage eingegangen sein, so wird der Empfang einer Multicast-Sammelanfrage weiter überwacht. Ist der erste Timer abgelaufen und keine Multicast-Sammelanfrage empfangen worden, so übernimmt der jeweilige Client die Aufgabe, selber eine Multicast-Sammelanfrage an einen ausgewählten Server zu richten. Dazu startet der jeweilige anfragende Client einen zweiten Timer (Timer 2) (Schritt 304) und übermittelt eine Multicast-Sammelanfrage an vorgebbare weitere Clients (Schritt 305). Nachfolgend überprüft der jeweilige anfragende Client, ob die Multicast-Sammelanfrage von den vorgebbaren weiteren Clients beantwortet wurde (Schritt 306), und ob der zweite Timer abgelaufen ist (Schritt 307). Sollte der zweite Timer noch nicht abgelaufen und keine Antwort auf die Multicast-Sammelanfrage eingegangen sein, so wird der Empfang einer Antwort auf die Multicast-Sammelanfrage weiter überwacht.

Nach Ablauf des zweiten Timers startet der jeweilige anfragende Client einen dritten Timer (Timer 3) (Schritt 308) und

- sendet eine Sammel-Verfügbarkeitsanfrage an den ausgewählten Server (Schritt 309). Die Sammel-Verfügbarkeitsanfrage umfaßt Daten des jeweiligen anfragenden Clients und derjenigen vorgebbaren weiteren Clients, die bis zum Ablauf des zweiten Timers auf die Multicast-Sammelanfrage des jeweiligen anfragenden Clients geantwortet haben. Nachfolgend überprüft der jeweilige anfragende Client, ob der ausgewählte Server Verfügbarkeit signalisiert hat (Schritt 310), und ob der dritte Timer abgelaufen ist (Schritt 311). Sollte der dritte Timer noch nicht abgelaufen und keine Antwort auf die Sammel-Verfügbarkeitsanfrage eingegangen sein, so wird der Empfang einer Antwort auf die Sammel-Verfügbarkeitsanfrage weiter überwacht.
- 15 Hat der ausgewählte Server seine Verfügbarkeit dem jeweiligen anfragenden Client signalisiert, so übermittelt der jeweilige anfragende Client eine positive Multicast-Verfügbarkeitsmeldung an die vorgebbaren weiteren Clients (Schritt 312). Sollte der ausgewählte Server seine Nichtverfügbarkeit signalisiert oder bis zum Ablauf des dritten Timers keine Antwort auf die Sammel-Verfügbarkeitsanfrage gesendet haben, so übermittelt der jeweilige anfragende Client eine negative Multicast-Verfügbarkeitsmeldung an die vorgebbaren weiteren Clients (Schritt 313).
- 25 Vorgebbare weitere Clients, die entsprechend der Überprüfung in Schritt 302 eine Multicast-Sammelanfrage empfangen haben, übermitteln ihre Keep-Alive-Daten an den jeweiligen anfragenden Client (Schritt 314), deren Empfang der jeweilige anfragende Client entsprechend Schritt 306 überprüft. Nachfolgend starten die vorgebbaren weiteren Clients einen vierten Timer (Timer 4) (Schritt 315) und überprüfen, ob eine positive oder negative Multicast-Verfügbarkeitsmeldung des jeweiligen anfragenden Clients empfangen wurde (Schritt 316), und ob der vierte Timer abgelaufen ist (Schritt 317). Bei Empfang einer Multicast-Verfügbarkeitsmeldung des jeweiligen anfragenden Clients wird der erste Timer zurückgesetzt (Schritt 318), so

13

daß die vorgebbaren weiteren Clients für eine durch den Ablauf des ersten Timers bestimmte Zeitdauer keine Verfügbarkeitsanfragen an den ausgewählten Server richten.

- 5 Solange der vierte Timer nicht abgelaufen ist, warten die vorgebbaren weiteren Clients auf eine Multicast-Verfügbarkeitsmeldung des jeweiligen anfragenden Clients. Sollte dies nicht vor Ablauf des vierten Timers geschehen, übernimmt der jeweilige Client nun selbst eine Rolle als Steller einer Sam-
- 10 mel-Verfügbarkeitsanfrage entsprechend den Schritten 304 bis 313, sofern der erste Timer abgelaufen ist bzw. vor seinem Ablauf keine weitere Multicast-Sammelanfrage eingeht.

- Die Anwendung der vorliegenden Erfindung ist nicht auf das
- 15 hier beschriebene Ausführungsbeispiel beschränkt.

## Patentansprüche

1. Verfahren zur Überprüfung der Verfügbarkeit eines Servers, bei dem
  - 5 - eine Verfügbarkeitsanfrage von einem Client an einen Server übermittelt wird,
  - die Verfügbarkeitsanfrage bei Verfügbarkeit des Servers durch eine vom Server an den Client übermittelte Bestätigungsmeldung beantwortet wird,
- 10 dadurch gekennzeichnet, daß der Client eine Meldung über die Verfügbarkeit des Servers an weitere Clients übermittelt, die daraufhin jeweils ein Übermitteln einer Verfügbarkeitsanfrage an den Server zumindest für ein vorgebbares Zeitintervall unterbinden.
- 15 2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß Daten zwischen dem Server und den Clients mittels verbindungsloser Vermittlungssteuerung übermittelt werden.
- 20 3. Verfahren nach einem der Ansprüche 1 oder 2, dadurch gekennzeichnet, daß die Meldung über die Verfügbarkeit des Servers an die vorgebbaren weiteren Clients mittels Multicast übermittelt wird.
- 25 4. Verfahren nach einem der Ansprüche 1 bis 3, dadurch gekennzeichnet, daß der Client nur vorgebbare weitere Clients innerhalb desselben Teilnetzes über die Verfügbarkeit des Servers informiert.
- 30 5. Verfahren nach einem der Ansprüche 1 bis 4, dadurch gekennzeichnet, daß der Client zu durch eine Zeitsteuerung vorgegebenen Zeitpunkten Verfügbarkeitsanfragen ausführt.

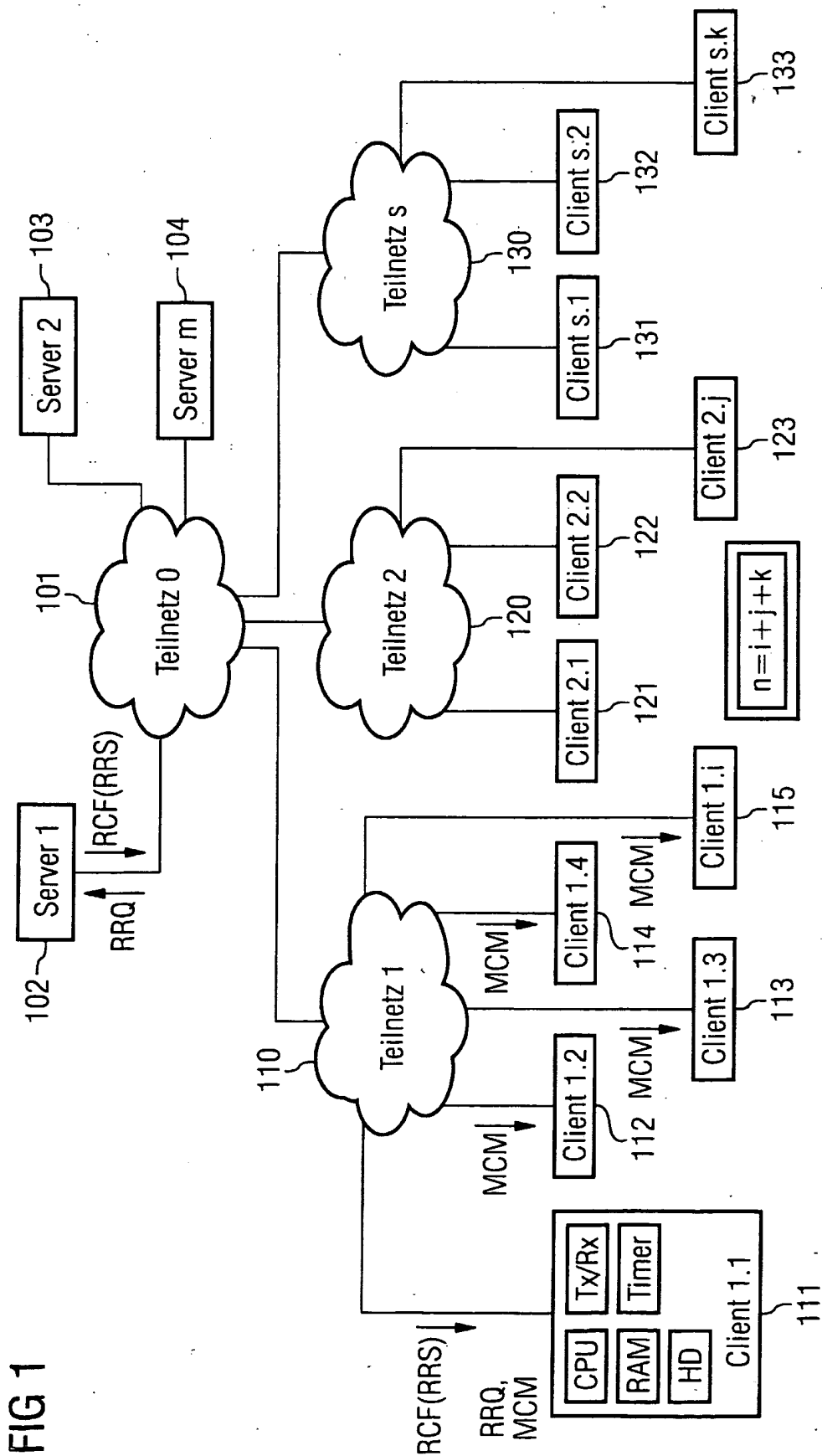
6. Verfahren nach Anspruch 5,  
dadurch gekennzeichnet, daß jeweils ein Zähler einer Zeit-  
steuerung, die einem vorgebbaren weiteren Client zugeordnet  
ist und Verfügbarkeitsanfragen veranlaßt, bei Empfang einer  
5 Meldung über die Verfügbarkeit des Servers auf einen vorgeb-  
baren Wert zurückgesetzt wird.
7. Steuerungsprogramm, das in einen Arbeitsspeicher eines  
Clients ladbar ist und zumindest einen Codeabschnitt auf-  
10 weist, bei dessen Ausführung
- eine Verfügbarkeitsanfrage von dem Client an einen Server  
übermittelt wird,
  - ein Empfang einer die Verfügbarkeitsanfrage bei Verfügbar-  
keit des Servers beantwortenden Bestätigungsmeldung über-  
15 wacht wird,
  - eine Meldung über die Verfügbarkeit des Servers an weitere  
Clients übermittelt wird,
  - ein Empfang einer Meldung eines vorgebbaren weiteren  
Clients über die Verfügbarkeit des Servers überwacht und  
20 ein Übermitteln einer Verfügbarkeitsanfrage an den Server  
zumindest für ein vorgebbares Zeitintervall bei Empfang  
einer solchen Meldung unterbunden wird,
- wenn das Steuerungsprogramm in dem Client abläuft.
- 25 8. Client für ein verbindungslose Dienste bereitstellendes  
Kommunikationsnetz mit
- einer Einrichtung zum Übermitteln einer Verfügbarkeitsan-  
frage an einen Server,
  - einer Einrichtung zur Überwachung eines Empfangs einer die  
30 Verfügbarkeitsanfrage bei Verfügbarkeit des Servers beant-  
wortenden Bestätigungsmeldung,
  - einer Einrichtung zur Übermittlung einer Meldung über die  
Verfügbarkeit des Servers an weitere Clients,
  - einer Einrichtung zur Überwachung eines Empfangs einer  
35 Meldung eines vorgebbaren weiteren Clients über die Ver-  
fügbarkeit des Servers und zum Unterbinden eines Übermit-  
telns einer Verfügbarkeitsanfrage an den Server zumindest

16

für ein vorgebbares Zeitintervall bei Empfang einer solchen Meldung.



1/3



2/3

FIG 2

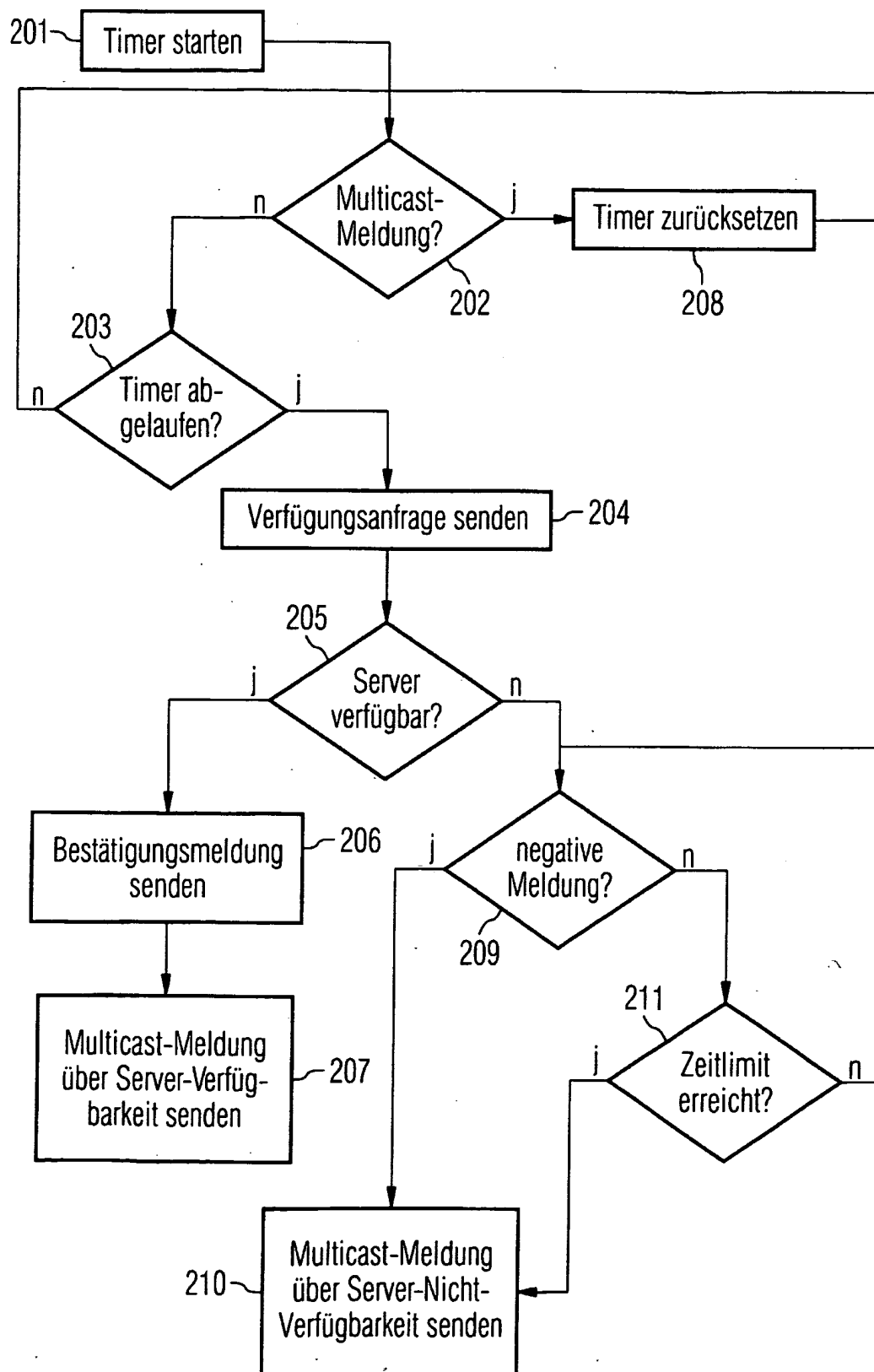


FIG 3

